M. ELIZABETH DAY (SBN 177125)
eday@feinday.com
DAVID ALBERTI (SBN 220265)
dalberti@feinday.com
SAL LIM (SBN 211836)
slim@feinday.com
MARC BELLOLI (SBN 244290)
mbelloli@feinday.com
FEINBERG DAY ALBERTI LIM & BELLOLI LLP
1600 El Camino Real, Suite 280
Menlo Park, CA 94025
Tel: 650.618.4360
Fax: 650.618.4368
Attorneys for Uniloc 2017 LLC

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

UNILOC 2017 LLC

Plaintiff,

v.

NETFLIX, INC.

Defendant.

CASE NO. 8:18-cv-02055

COMPLAINT FOR PATENT INFRINGEMENT

DEMAND FOR JURY TRIAL

Plaintiff Uniloc 2017 LLC ("Uniloc"), by and through the undersigned counsel, hereby files this Complaint and makes the following allegations of patent infringement relating to U.S. Patent Nos. 8,407,609, 6,584,229 and 6,519,005 against Defendant Netflix, Inc. ("Netflix"), and alleges as follows upon actual knowledge with respect to itself and its own acts and upon information and belief as to all other matters:

NATURE OF THE ACTION

1. This is an action for patent infringement. Uniloc alleges that Netflix infringes U.S. Patent Nos. 8,407,609 (the "'609 patent"), 6,584,229 (the "'229 patent") and 6,519,005 (the "'005 patent"), copies of which are attached hereto as Exhibits A-C (collectively, "the Asserted Patents").

2. Uniloc alleges that Netflix directly infringes the Asserted Patents by making, using, offering for sale, selling, licensing and/or importing products and services that: (1) track digital media presentations such as Netflix Service, (2) perform coding of image signals using the VP9 codec and (3) perform a method for motion coding an uncompressed (pixel level) digital video data stream. Uniloc seeks damages and other relief for Netflix's infringement of the Asserted Patents.

THE PARTIES

3. Uniloc 2017 LLC is a Delaware corporation having places of business at 1209 Orange Street, Wilmington, Delaware 19801 and 620 Newport Center Drive, Newport Beach, California 92660.

4. Uniloc holds all substantial rights, title and interest in and to the Asserted Patents.

5. Upon information and belief, Defendant Netflix, is a corporation organized and existing under the laws of the State of Delaware, with at least the following places of business in this District: 5808 Sunset Blvd, Los Angeles, CA 90028 and 335 N. Maple Dr. Beverly Hills, CA 90210. Netflix can be served with

process by serving its registered agent for service of process in California: CT Corporation System, 818 W. Seventh St, Suite 930, Los Angeles, CA. 90017.

JURISDICTION AND VENUE

6. This action for patent infringement arises under the Patent Laws of the United States, 35 U.S.C. § 1 et. seq. This Court has original jurisdiction under 28 U.S.C. §§ 1331 and 1338.

7. This Court has both general and specific jurisdiction over Netflix because Netflix has committed acts within the Central District of California giving rise to this action and has established minimum contacts with this forum such that the exercise of jurisdiction over Netflix would not offend traditional notions of fair play and substantial justice. Defendant Netflix, directly and through subsidiaries, intermediaries (including distributors, retailers, franchisees and others), has committed and continues to commit acts of patent infringement in this District, by, among other things, making, using, testing, selling, licensing, importing and/or offering for sale/license products and services that infringe the Asserted Patents.

8. Venue is proper in this district and division under 28 U.S.C. §§ 1391(b)-(d) and 1400(b) because Netflix has committed acts of infringement in the Central District of California and has multiple regular and established places of business in the Central District of California.

COUNT I – INFRINGEMENT OF U.S. PATENT NO. 8,407,609

9. The allegations of paragraphs 1-8 of this Complaint are incorporated by reference as though fully set forth herein.

10. The ’609 patent, titled “System and Method For Providing And Tracking The Provision of Audio and Visual Presentations Via A Computer Network,” issued on March 26, 2013. A copy of the ’609 patent is attached as Exhibit A.

11. Pursuant to 35 U.S.C. § 282, the ’609 patent is presumed valid.

12. Invented by LINQware, Inc., the inventions of the '609 patent were not well-understood, routine or conventional at the time of the invention. At the time of invention of the '609 patent, it was very difficult for a user of an Internet enabled computer to find content of a particular type and relating to a particular subject because the amount of content available via the Internet was virtually unlimited. '609 patent at 1:50-54. A popular solution to finding desired content was to use a publicly available search engine. *Id.* at 1:55-56. Each search engine typically used its own methodology to create indices such that, ideally, only meaningful results are returned for each query. *Id.* at 1:62-64. This was not always true though due to the complex nature and nuances of human language and efforts by document authors or providers to fool or trick the indexer into ranking its documents above those of others. *Id.* at 1:64-2:1. Examples of conventional search engines include those made available via www.yahoo.com, www.google.com and www.search.com. *Id.* at 2:1-3.

13. The inventive solution of the claimed inventions of the '609 patent provides a method whereby digital media presentations are delivered and tracked from a first computer system to a user's computer via a network in a manner that departs from convention. *Id.* at 2:13-15. In accordance with one aspect of the invention, a web page, identifier data and an applet are provided to the user's computer for each digital media presentation to be delivered using the first computer system. *Id.* at 2:15-21. The applet is operative by the user's computer as a timer. *Id.* at 2:21-22. The first computer system receives at least a portion of the identifier data from the user's computer responsively to the timer applet each time a predetermined temporal period elapses and stores data indicative of the received at least portion of the identifier data. *Id.* at 2:22-27. Each provided webpage causes corresponding digital media presentation data to be streamed from a second computer system distinct from the first computer system directly to the user's

computer independent of the first computer system. *Id.* at 2:27-31. The stored data is indicative of an amount of time the digital media presentation data is streamed from the second computer system to the user's computer. *Id.* at 2:31-34.

14. A person of ordinary skill in the art reading the '609 patent and its claims would understand that the patent's disclosure and claims are drawn to solving a specific, technical problem arising in delivering and tracking digital media presentations via a network. Moreover, a person of ordinary skill in the art would understand that the claimed subject matter of the '609 patent presents advancements in the field of the provision of informational, entertainment, educational, business and other audio and/or audio/visual presentations via a computer network. And, as detailed by the specification, the prior search engines suffered drawbacks such that a new and novel communications system was required.

15. In light of the foregoing, a person of ordinary skill in the art would understand that claim 1 of the '609 patent is directed to a specific method for providing and tracking digital media presentations using a web page, identifier data and a timer applet originating at a first computer to track and responsively stream a digital media presentation from a second computer that can be viewed by a user at the user's computer. *Id.* at 14:17-45. Moreover, a person of ordinary skill in the art would understand that claim 1 of the '609 patent contains the inventive concept of providing and tracking digital media presentations using a web page, identifier data and a timer applet originating at a first computer to track and responsively stream a digital media presentation from a second computer that can be viewed by a user at the user's computer. *Id.*

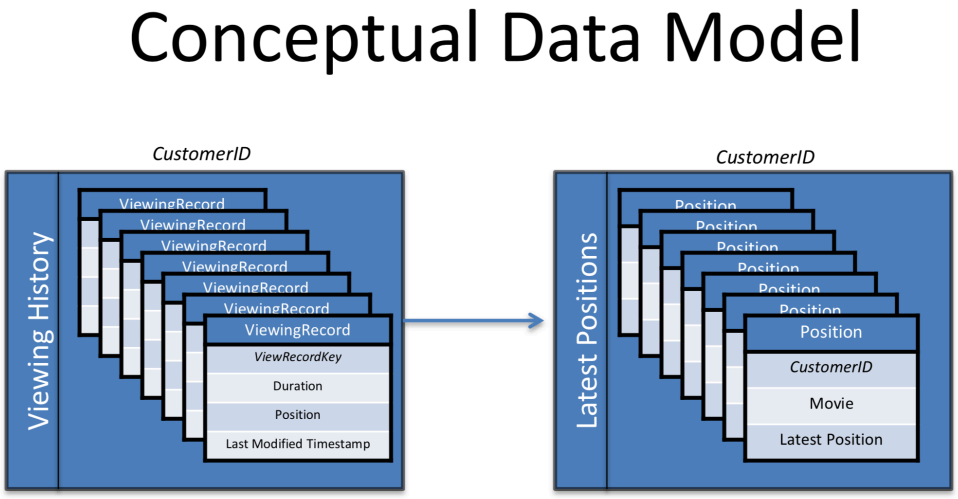
16. Upon information and belief, Netflix makes, uses, offers for sale, sells in the United States, licenses in the United States and/or imports into the United States Netflix Service which provides a method for tracking digital media

presentations (collectively the "Accused Infringing Devices").

17. Upon information and belief, the Accused Infringing Devices infringe at least claim 1 in the exemplary manner described below.

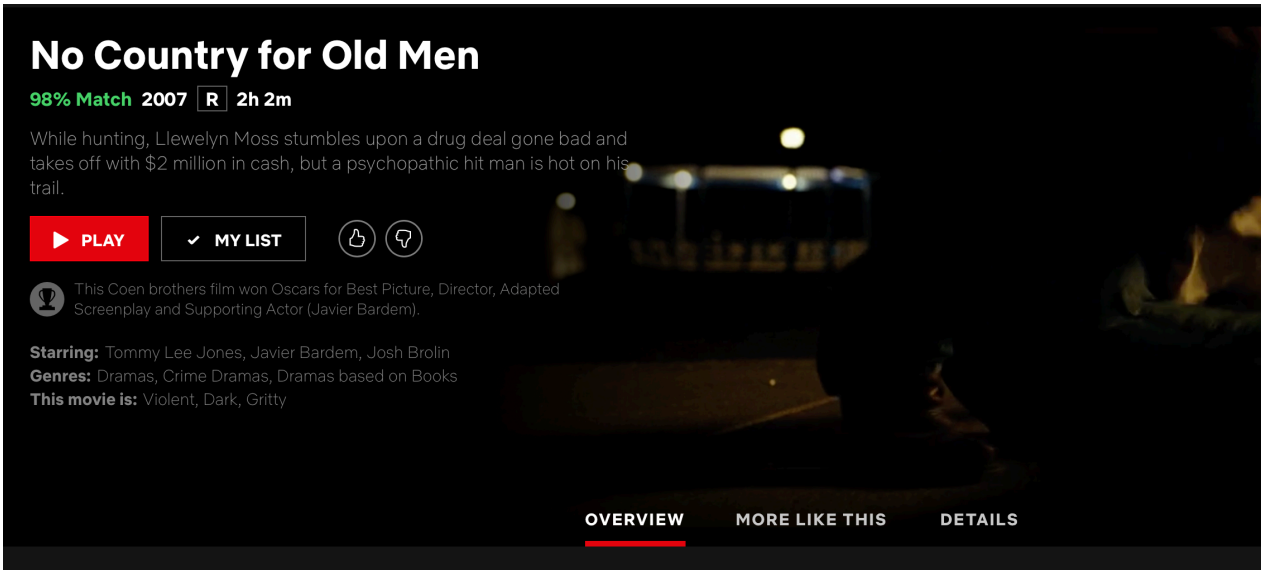
18. The Accused Infringing Devices track digital media presentations delivered from a first computer system to a user's computer via a network.

Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 9.



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 30.

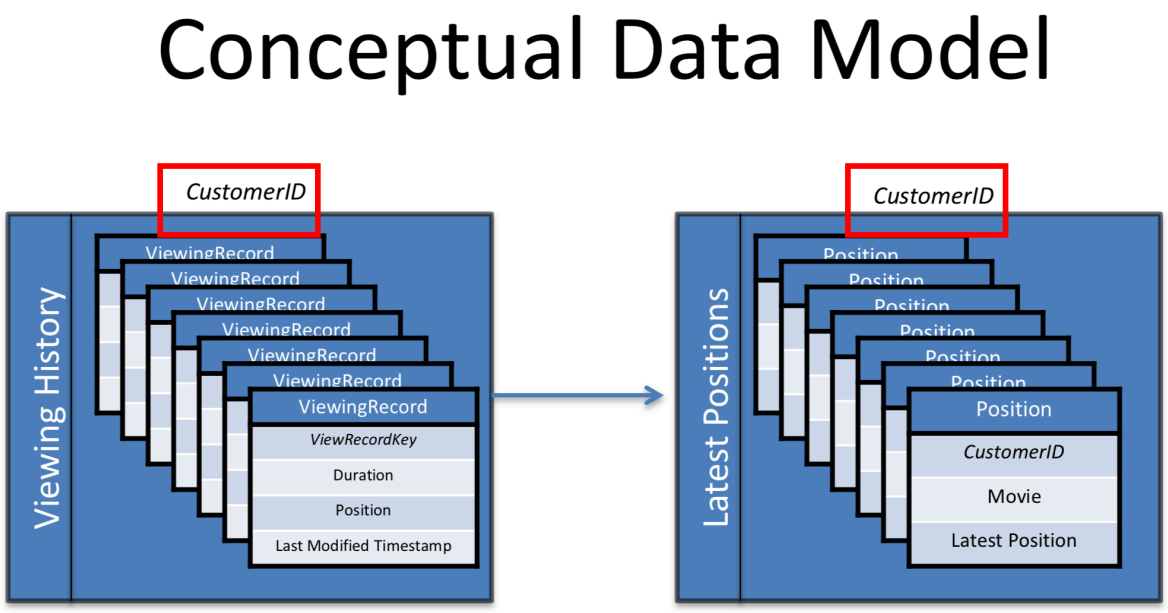
19. The Accused Infringing Devices provide a corresponding web page to the user's computer for each digital media presentation to be delivered using the first computer system.



Source: Screenshot

20. The Accused Infringing Devices provide identifier data to the user's computer using the first computer system.

Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 9.

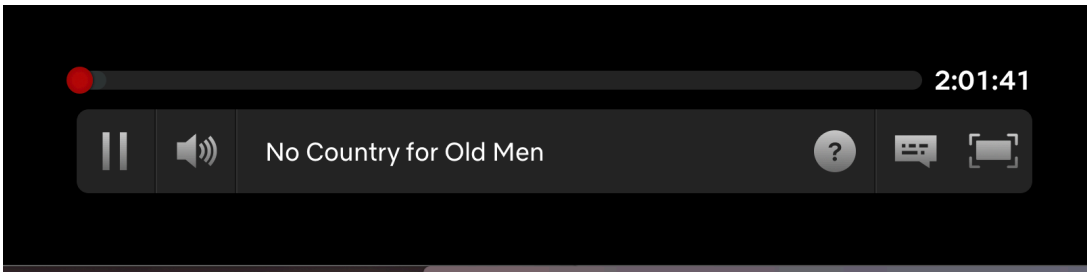


Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 30

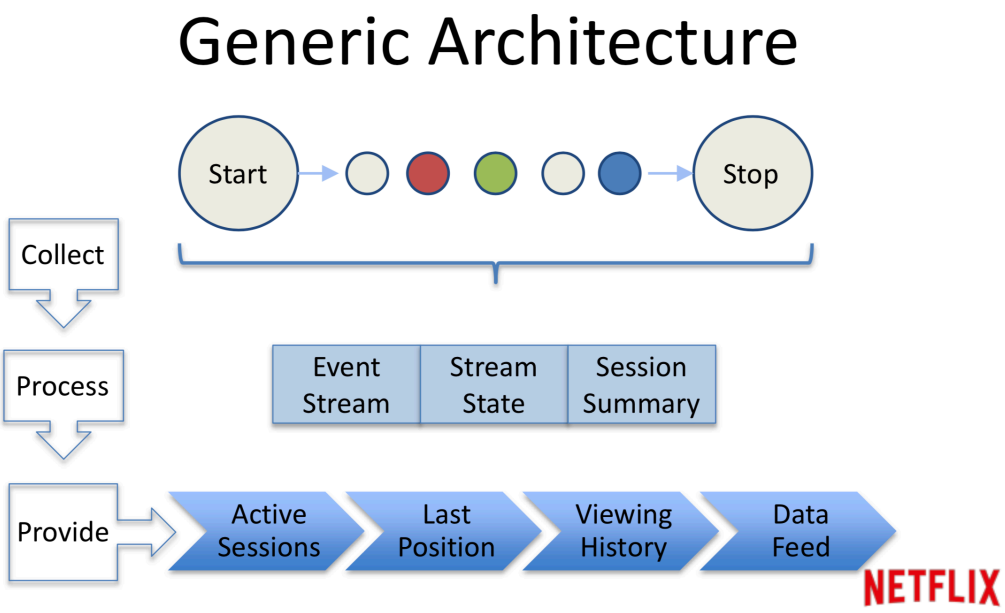
21. The Accused Infringing Devices provide an applet to the user's computer for each digital media presentation to be delivered using the first computer system. In particular, the Accused Infringing Devices provide a script that keeps track of how much of the presentation the user has watched, thus reflecting the operation of a time running in the background.

```
C.r("akira/components/controls/Progress/Progress.jsx", function(r, s, e) {
    "use strict";
    var t = r("prop-types"),
        n = r("react"),
        o = r("create-react-class"),
        a = o({
            displayName: "Progress",
            contextTypes: {
                getI18nString: t.func.isRequired
            },
            render: function() {
                var r = this.props.variants,
                    s = parseInt(this.props.bookmarkPosition / 60, 10),
                    e = parseInt(this.props.runtime / 60, 10);
                if (!this.props.runtime)
                    return null;
                var t = {
                        width: parseInt(100 * this.props.bookmarkPosition / this.props.runtime, 10) + "%"
                    },
                    o = r ? r.map(function(r) {
                        return "progress--" + r
                    }).join(" ") : "";
                return n.createElement("div", {
                    className: "progress " + o
                }, n.createElement("span", {
                    className: "progress-bar"
                }, n.createElement("span", {
                    role: "presentation",
                    className: "progress-completed",
                    style: t
                })), this.props.showSummary ? n.createElement("span", {
                    className: "summary"
                }, this.context.getI18nString("discovery/akira/Common", "progress.summary", {
                    BOOKMARK: s,
                    RUNTIME: e
                })) : null)
            }
        });
    s.exports = a
});
```

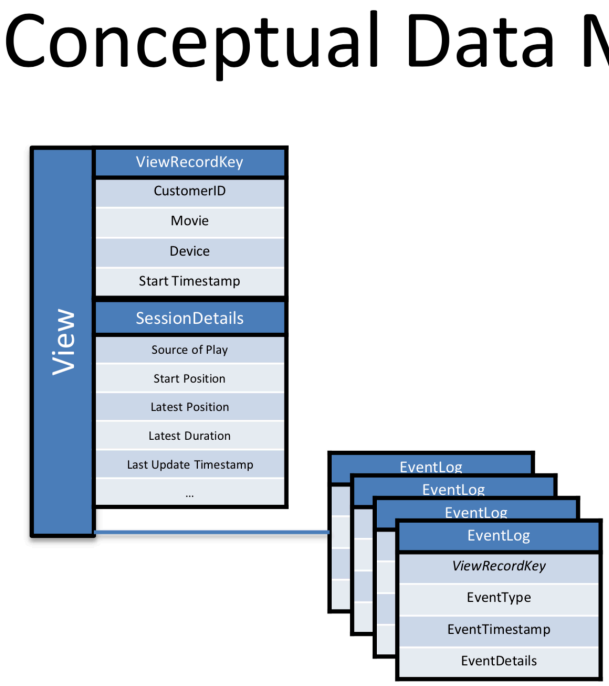
Source: Script from http://codex.nflxext.com.



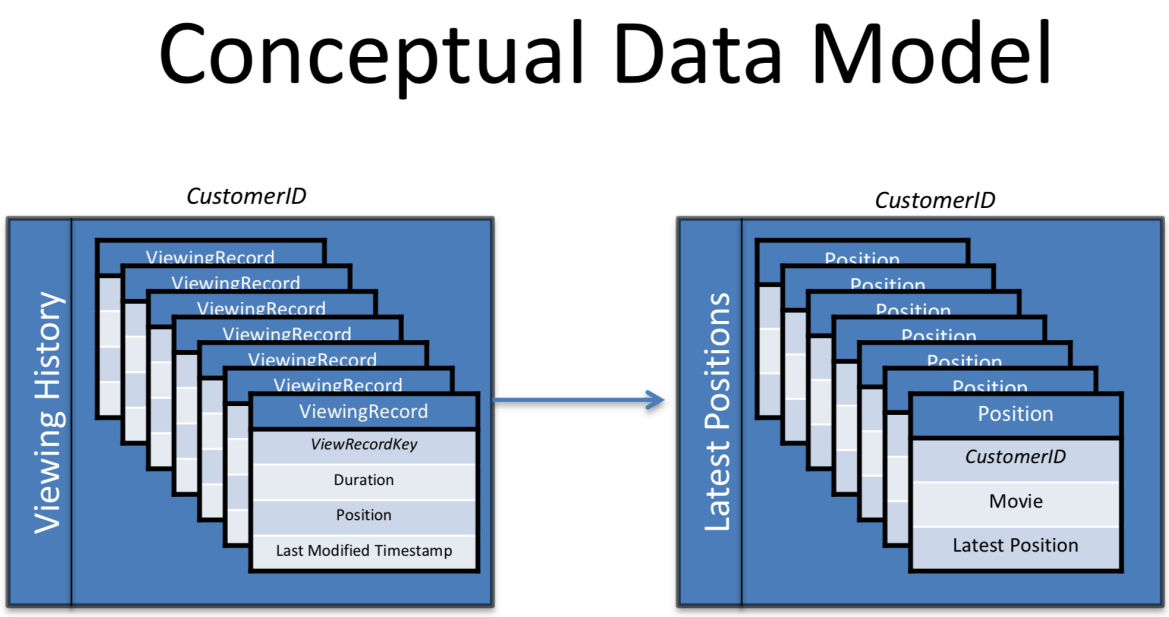
Source: Screenshot



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 24.

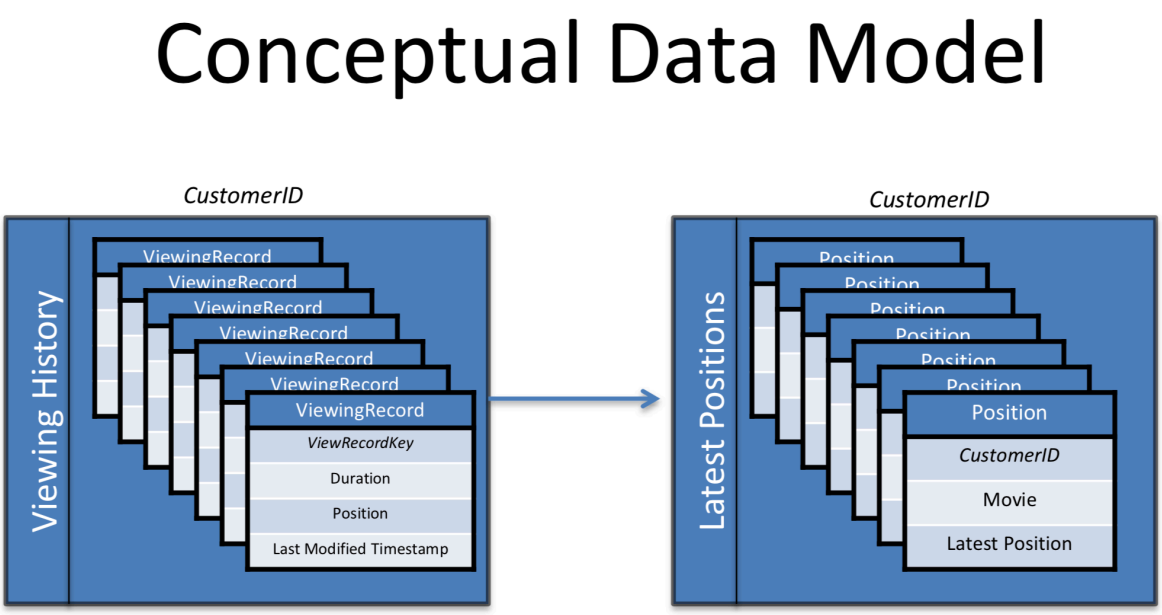


Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 29.



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 30.

22. The Accused Infringing Devices receive at least a portion of the identifier data from the user's computer responsively to the timer applet each time a predetermined temporal period elapses using the first computer system. The Accused Infringing Devices maintain a viewing history for each user, shown below as a collection of consecutive viewing records. Each viewing record includes a ViewRecordKey to identify a program as well as the duration, position and last modified timestamp at the time the viewing record was updated. The viewing record is updated periodically, at least when a heartbeat is sent reflecting operation of a timer.



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 30.

23. In addition, the Accused Infringing Devices' player periodically sends a heartbeat to the server to indicate the current playback position in the event the user needs to return to the current position in the event of an error or crash. The heartbeat is associated with the active session, which by definition, must identify the user. The heartbeat is sent at predetermined intervals. The Accused Infringing Devices also take snapshots of the viewing history, including positions, at intervals.

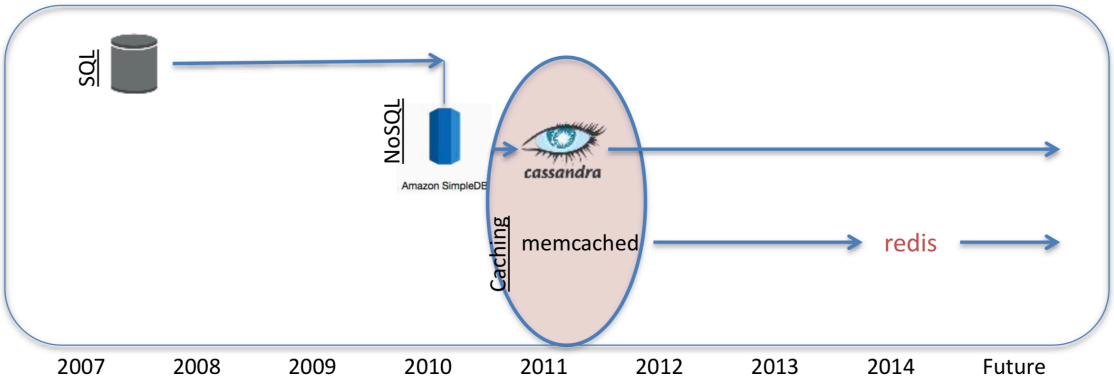
Command Use Cases

Action	Operation	Key	DataSet
Start	Insert	ViewRecordKey	ActiveSession ViewingRecord
Continue (heartbeat)	Update	ViewRecordKey	ActiveSession
Log	Insert	ViewRecordKey	EventLog
Stop	Update	ViewRecordKey	ActiveSession ViewingRecord
Snapshot	Insert/Update	CustomerID	ViewingHistory Positions

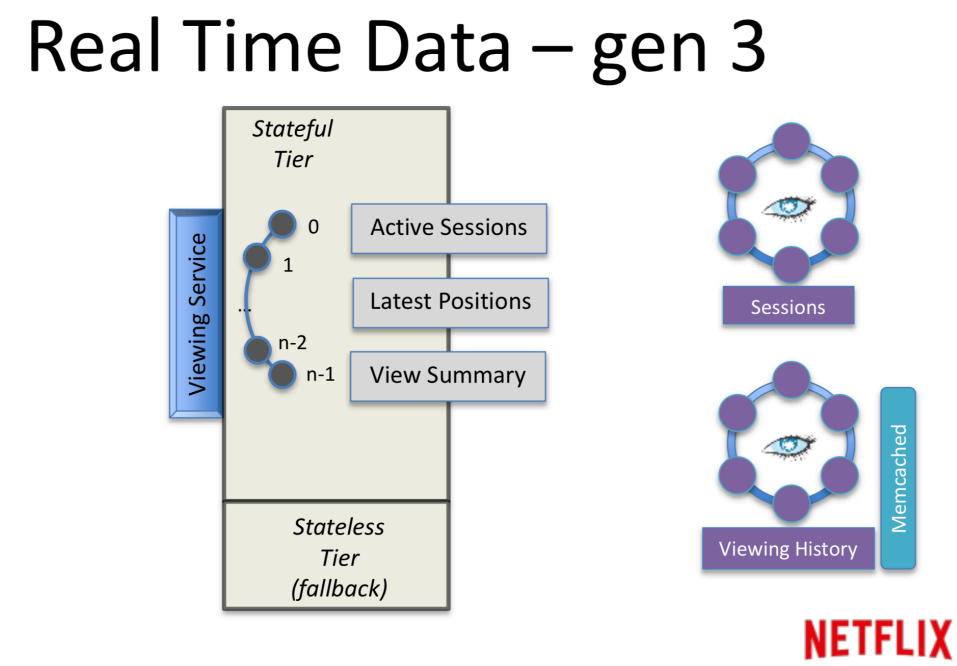
Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 31.

24. The Accused Infringing Devices store data indicative of the received at least portion of the identifier data using the first computer system. The user's viewing history, updated every time a heartbeat is sent, is stored by the Accused Infringing Devices using one of Cassandra, the Viewing History Tier or memcached, as shown below.

Real Time Data – gen 3



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 42.



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 45.

gen 3 – Cluster Scale

Cluster	Scale
Cassandra Viewing History	~100 hi1.4xl nodes ~48 TB total space used
Viewing Service Stateful Tier	~1700 r3.2xl nodes 50GB heap memory per node
Memcached	~450 r3.2xl/xl nodes ~8TB memory used

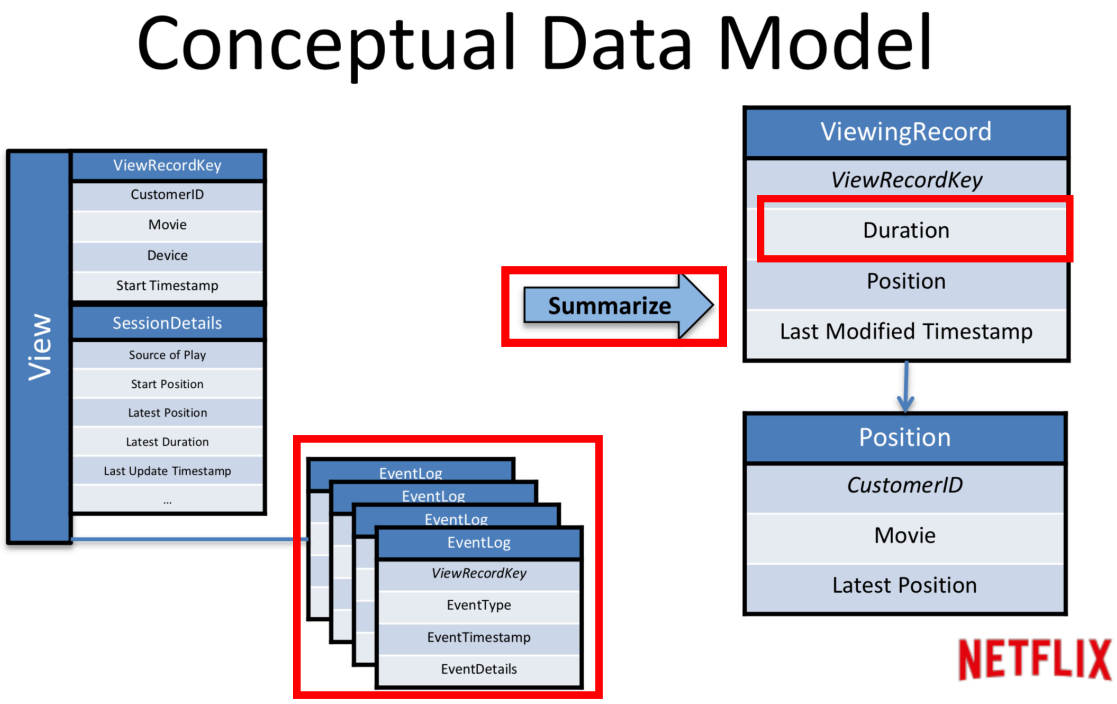
Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 59.

25. Each provided webpage causes corresponding digital media presentation data to be streamed from a second computer system (e.g., the content delivery network, *e.g.*, NOpen Connect), distinct from the user's computer independent of the first computer system (e.g., the Netflix website).

> Netflix Open Connect delivers 100% of our video traffic, currently *over 125 million hours of viewing per day*. This amounts to tens of terabits per second of simultaneous peak traffic, making Netflix Open Connect one of the highest-volume networks in the world.
>
> Globally, close to 90% of our traffic is delivered via direct connections between Open Connect and the residential Internet Service Providers (ISPs) our members use to access the internet. Most of these connections are localized to the regional point of interconnection that's geographically closest to the member who's watching. Because connections to the Netflix Open Connect network are always free and our traffic delivery is highly localized, thousands of ISPs around the world enthusiastically participate.
>
> We also give qualifying ISPs the same Open Connect Appliances (OCAs) that we use in our internet interconnection locations. After these appliances are installed in an ISP's data center, almost all Netflix content is served from the local OCAs rather than "upstream" from the internet. Many ISPs take advantage of this option, in addition to local network interconnection, because it reduces the amount of capacity they need to build to the rest of the internet since Netflix is no longer a significant factor in that capacity. This has the dual benefit of reducing the ISP's cost of operation and ensuring the best possible Netflix experience for their subscribers.
>
> We now have Open Connect Appliances in close to 1,000 separate locations around the world. In big cities like New York, Paris, London, Hong Kong, and Tokyo, as well as more remote locations — as far north as Greenland and Tromsø, Norway and as far south as Puerto Montt, Chile, and Hobart, Tasmania. ISPs have even placed OCAs in Macapá and Manaus in the Amazon rainforest — on every continent, except Antarctica and on many islands such as Jamaica, Malta, Guam, and Okinawa. This means that most of our members are getting their Netflix audio and video bits from a server that's either inside of, or directly connected to, their ISP's network within their local region.

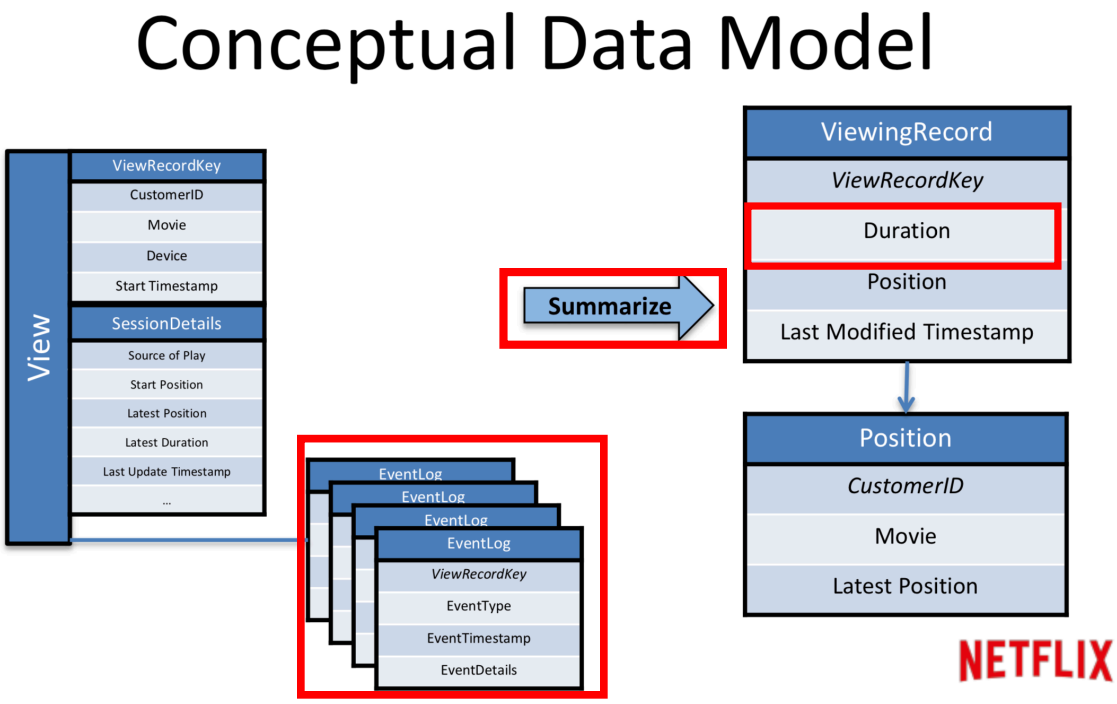
Source: https://media.netflix.com/en/company-blog/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience.

26. The stored data is indicative of the amount of time the digital media presentation is streamed from the second computer system to the user's computer. The stored data indicates the duration and position of the user's current position, which indicates the amount of time the presentation has been streamed to the user's computer by the CDN.



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 29.

27. Each stored data is together indicative of a cumulative time the corresponding web page was displayed by the user's computer. The user visits netflix.com, which displays tiles and information screens for movies and TV shows. When the user elects to stream a movie or TV show, the player is loaded on the same page. The amount of time the user spends watching a movie or TV show is tracked by Netflix and also reflects the amount of time the preceding webpage was displayed by the user's computer.



Source: https://www.slideshare.net/PhilipFisherOgden/netflix-viewing-history-ebjug-2014 at 29.

28. Netflix has infringed, and continues to infringe, at least claim 1 of the '609 patent in the United States, by making, using, offering for sale, selling, licensing and/or importing the Accused Infringing Devices in violation of 35 U.S.C. § 271(a).

29. Upon information and belief, Netflix may have infringed and continues to infringe the '609 patent through other software and devices utilizing the same or reasonably similar functionality, including other versions of the Accused Infringing Devices.

30. Netflix's acts of direct infringement have caused and continue to cause damage to Uniloc and Uniloc is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

COUNT II – INFRINGEMENT OF U.S. PATENT NO. 6,584,229

31. The allegations of paragraphs 1-8 of this Complaint are incorporated by reference as though fully set forth herein.

32. The '229 patent, titled "Macroblock-Based Object-Oriented Coding

Method Of Image Sequence Having A Stationary Background," issued on June 24, 2003. A copy of the '229 patent is attached as Exhibit B.

33. Pursuant to 35 U.S.C. § 282, the '229 patent is presumed valid.

34. Invented by Electronics and Telecommunications Research Institute, the inventions of the '229 patent were not well-understood, routine or conventional at the time of the invention. At the time of invention of the '229 patent, video compression techniques existed for reducing the size of information using object-based coding techniques, where an image is divided into a background region and a foreground object region and each region is encoded separately. '229 patent at 1:21-25. According to the prior art, each divided region is coded using a conventional macro block-based image coding technique, such as discrete cosine transformation and quantization techniques. *Id*. at 1:25-34. In addition, shape information representing each divided region is separately coded using macroblocks and transmitted in order to reconstruct the image signal at the receiving end. *Id*. at 1:34-37.

35. In these conventional video compression methods, it is important to efficiently divide the image signal into two regions and to code the shape information of each divided region at a very high compression rate in order to maximize the coding efficiency. *Id*. at 1:38-41. A conventional region division technique can be used to divide the image signal in a pixel unit by use of temporal change information or spatial change information of the image signal. *Id*. at 1:41-44. Very complicated processes are executed in order to divide the video by pixel unit, which are difficult to embody in real time. *Id*. at 1:50-54. In addition, the prior art techniques may reach a limitation to maximize the coding efficiency, because lots of bits are consumed in order to code and transmit the shape information of each region acquired by pixel unit. *Id*. at 1:54-58. The coding rate of the shape information could not be lowered below a certain level, even though a

total coding rate is lowered. *Id*. at 1:58-60. Therefore, these conventional techniques were not efficiently utilized to code at a desired low bit rate. *Id*. at 1:60-63.

36. The invention of the ’229 patent addresses these drawbacks by providing a macroblock-based object-oriented coding method of image sequence having a stationary background, which can be coded at a low coding bit rate. ’229 patent at 1:66-2:8. The method codes the shape information of each region with the help of dividing a video not by pixel unit but by macroblock-unit, and with the help of dividing the video into two regions and coding each region in order to solve the above-mentioned problems. *Id*.

37. A person of ordinary skill in the art reading the ’229 patent and its claims would understand that the patent’s disclosure and claims are drawn to solving a specific, technical problem arising in in the field of video compression. Moreover, a person of ordinary skill in the art would understand that the claimed subject matter of the ’229 patent presents advancements in the field of video compression. And, as detailed by the specification, the prior compression techniques suffered drawbacks such that a new and novel communications system was required.

38. In light of the foregoing, a person of ordinary skill in the art would understand that claim 1 of the ’229 patent is directed to a method for macroblock-based object oriented coding of an image signal having a stationary background region and an object region, wherein coded macroblock information from a previous frame is reused when a difference between a pixel value of the macroblock in a current frame and previous frame is equal to or below a threshold value. Moreover, a person of ordinary skill in the art would understand that claim 1 of the ’229 patent contains that corresponding inventive concept.

39. Upon information and belief, Netflix makes, uses, offers for sale, sells

in the United States, licenses in the United States and/or imports into the United States products and services that use VP9 video encoding for coding image signals (collectively the "Accused Infringing Devices").

40. Upon information and belief, the Accused Infringing Devices infringe at least claim 1 in the exemplary manner described below.

41. The Accused Infringing Devices practice a method for use in a macro-block-based object oriented coding of an image signal using the VP9 codec wherein the image signal has a stationary background region and an object region and contains a current frame and a previous frame, each frame including a plurality of macroblocks.

> **What's new about our mobile encodes**
>
> We are introducing two new types of mobile encodes—AVCHi-Mobile and VP9-Mobile. The enhancements in the new bitstreams fall into three categories: (1) new video compression formats, (2) more optimal encoder settings, and (3) per-chunk bitrate optimization. All the changes combined result in better video quality for the same bitrate compared to our current streams (AVCMain).

Source: https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909

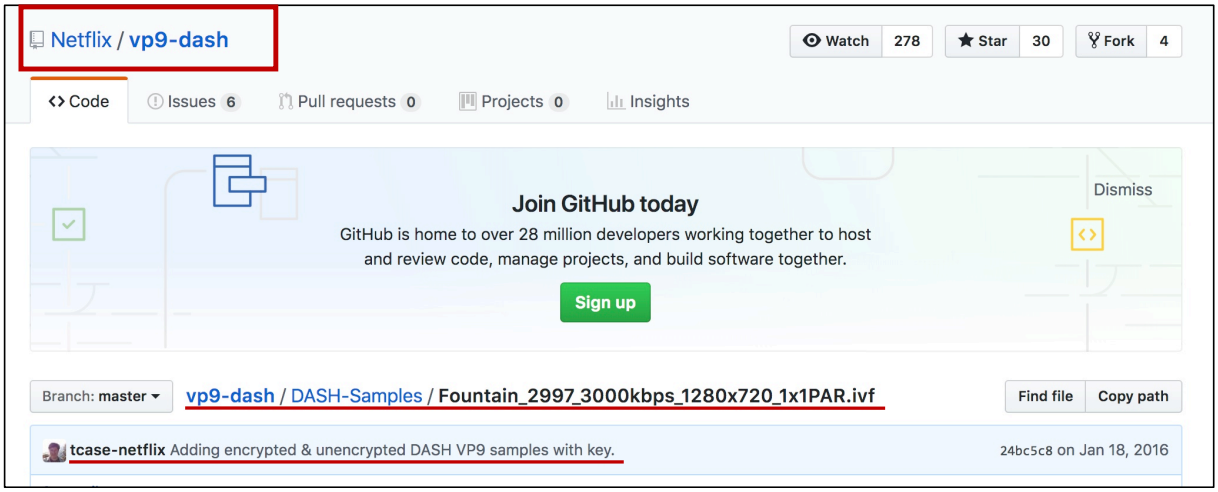
> VP9 has a number of tools which bring improvements in compression efficiency over H.264/AVC, including:
>
> - Motion-predicted blocks of sizes up to 64×64
> - ⅛th pel motion vectors
> - Three switchable 8-tap subpixel interpolation filters
> - Better coding of motion vectors
> - Larger discrete cosine transforms (DCT, 16×16, and 32×32)
> - Asymmetric discrete sine transform (ADST)
> - Loop filtering adapted to new block sizes
> - Segmentation maps

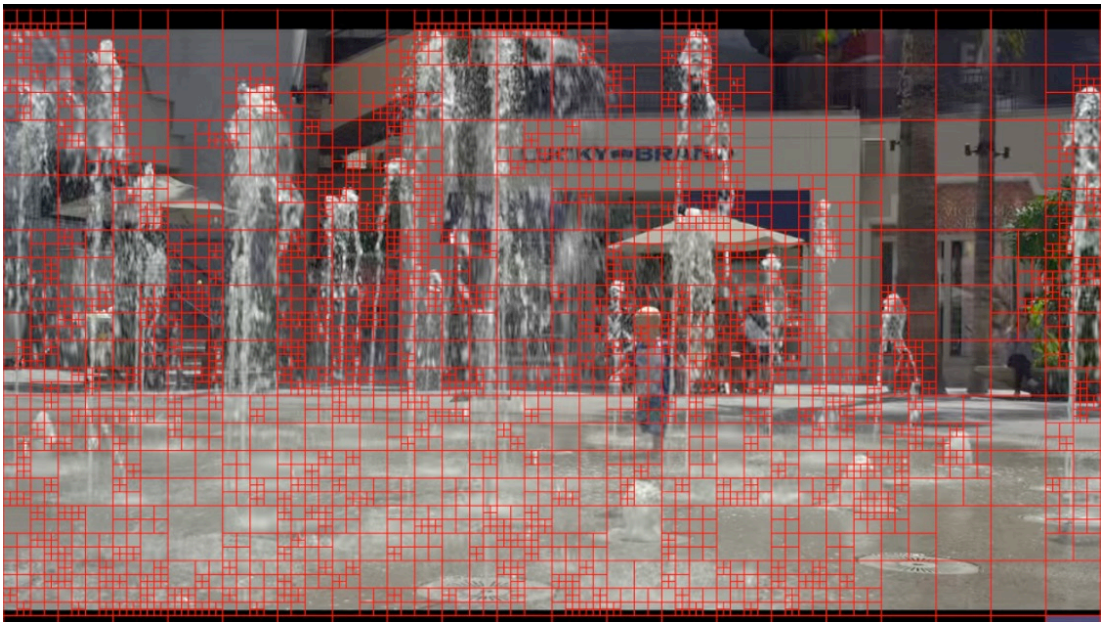
Source: https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909

> **2.4. Transforms**
>
> VP9 supports the Discrete Cosine Transform (DCTs) at sizes 4x4, 8x8, 16x16 and 32x32 and removes the second-order transform that was employed in VP8. Only transform sizes equal to, or smaller than, the prediction block size may be specified. Modes B_PRED and 4x4 SPLITMV are thus restricted to using only the 4x4 transform; modes I8X8_PRED and non-4x4 SPLITMV can use either the 4x4 or 8x8 transform; full-size (16x16) macroblock predictors can be coupled with either the 4x4, 8x8 or 16x16 transforms, and superblocks can use any transform size up to 32x32. Further restrictions on the available sub-set of transforms can be signaled at the frame-level, by specifying a maximum allowable transform size, or at the macroblock level by explicitly signaling which of the available transform sizes is used.

Source: https://tools.ietf.org/id/draft-grange-vp9-bitstream-00.html#rfc.section.2.1

42. Netflix supplies VP9 sample test files to developers. Those samples demonstrate the block based coding of video images.

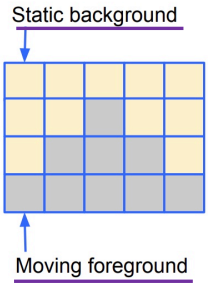




Source: https://github.com/Netflix/vp9-dash/blob/master/DASH-Samples/Fountain_2997_3000kbps_1280x720_1x1PAR.ivf

43. VP9 coding uses segmentation to group together blocks with common characteristics, including static background and moving foreground (object) regions.

- Segmentation feature significantly enhanced in VP9
  - Groups together blocks that share common characteristics into segments.
  - Indicate segmentation id at block level
    - Differentially encode segmentation map temporally

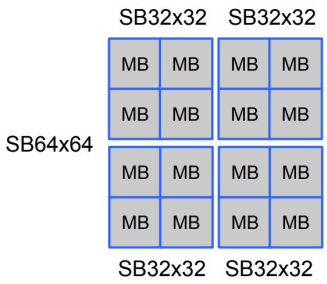


Source: https://files.meetup.com/9842252/Overview-VP9.pdf

44. The Accused Infringing Devices use inter mode prediction at a block level to predict a current block relative to a reference frame. The frame is coded using a variety of block sizes and a plurality of blocks to encode the image.

- Information conveyed at block end-points:
  - Prediction mode (can be INTER or INTRA)
  - If INTER mode:
    - Prediction reference frame(s)
    - Motion vector(s) if needed
    - Sub-pel filter choice
  - Skip flag indicating if there is any non-zero coefficient for prediction residual

- VP9 introduces Superblocks (SB*mxn*):
  - SB64x64: 64x64 block
  - SB64x32: 64x32 block
  - ...
  - MB16x16: 16x16 block
  - ...
  - SB8x8: 8x8 block

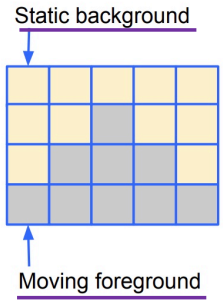


Source: https://files.meetup.com/9842252/Overview-VP9.pdf

45. The Accused Infringing Devices use VP9 segmentation to divide the inputted video into a stationary background and the object region on a block by block basis to create segmentation maps. For INTER blocks, VP9 uses the

calculation of motion vectors temporally between frames to find a difference (residual) between blocks in a given frame (next frame) and a reference frame.

- Segmentation feature significantly enhanced in VP9
  - Groups together blocks that share common characteristics into segments.
  - Indicate segmentation id at block level
    - Differentially encode segmentation map temporally



Source: https://files.meetup.com/9842252/Overview-VP9.pdf

How Does it Work?

While 4K video increases picture quality by making individual pixels smaller, VP9 codec and HEVC make them bigger to reduce the bitrate and file size. While this may seem conflicting, the encoding engine takes the larger pixels and turns them into a higher resolution output. Source video, consisting of video frames, is encoded or compressed to create a compressed video bitstream. Each individual frame is first broken up into blocks of pixels. The blocks are then analyzed for spatial redundancies and temporal linkages between frames are analyzed to take advantage of areas that do not change. These are encoded via motion vectors that predict qualities of the given block on the next frame. The residual information is encoded using an efficient binary compression.

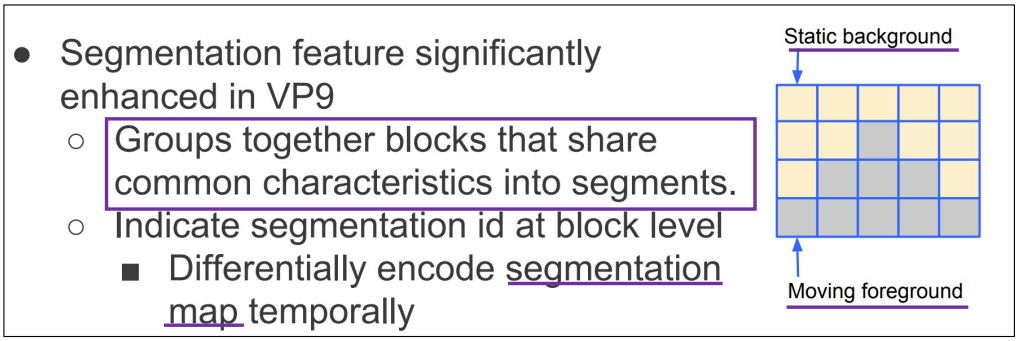
Source: https://www.encoding.com/vp9/

- Information conveyed at block end-points:
  - Prediction mode (can be INTER or INTRA)
  - If INTER mode:
    - Prediction reference frame(s)
    - Motion vector(s) if needed
    - Sub-pel filter choice
  - Skip flag indicating if there is any non-zero coefficient for prediction residual

Source: https://files.meetup.com/9842252/Overview-VP9.pdf

46. The Accused Infringing Devices code macroblocks into segments

(regions) on a block by block basis to create segmentation maps that code the shape information.



Source: https://files.meetup.com/9842252/Overview-VP9.pdf

Segment ID by Blocksize Hide

	64x64	32x32	16x16	8x8	4x4
0	13	37	49	16	0
1	6	7	52	52	0
2	12	42	219	384	0
3	26	124	461	720	0
4	5	35	325	916	0
5	0	0	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0

Source: Frame from the sample file showing VP9 image blocks using a VP9 analyzer showing the use of blocks and segments.

47. The Accused Infringing Devices use the well-known coding technique of motion vectors, prediction residuals and transform coding to code the pixel information of blocks in an object region by using, for example, INTER block coding.

- Information conveyed at block end-points:
  - Prediction mode (can be INTER or INTRA)
  - If INTER mode:
    - Prediction reference frame(s)
    - Motion vector(s) if needed
    - Sub-pel filter choice
  - Skip flag indicating if there is any non-zero coefficient for prediction residual

- VP9 uses different transforms for different modes:
  - 2D DCT for INTER modes and
  - Hybrid DCT/ADST transforms for INTRA modes
  - A lossless 4x4 transform for lossless encoding

Source: https://files.meetup.com/9842252/Overview-VP9.pdf

48. The Accused Infringing Devices use segments that are background (static) regions of inter blocks which contain blocks that are encoded from a fixed reference (previous) frame. The coded pixel information for a block contains the residual between the previous frame and the current frame, which can be zero (skip), using the coded pixel information of a previous frame block corresponding to a current frame block (only a motion vector pointing to the reference frame).

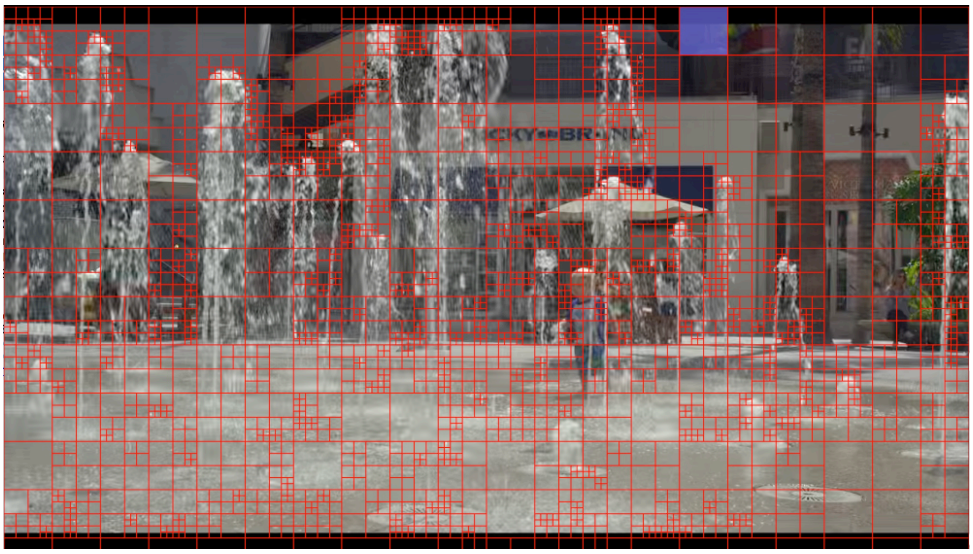
- the segment ID allows selecting a per-block quantizer and/or loop-filter strength level that is different from the frame-wide default, which allows for adaptive quantization/loop-filtering. The segment ID also allows encoding a fixed reference and/or marking a block as skipped, which is mainly useful for static content/background;

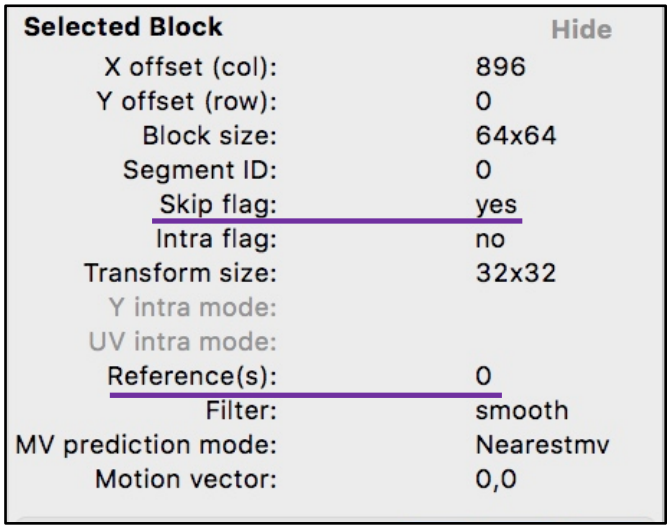
Source: https://files.meetup.com/9842252/Overview-VP9.pdf

Txsize by Blocksize (inter)					Hide
	64x64	32x32	16x16	8x8	4x4
Skip	40	112	610	1441	0
32x32	3	77			
16x16	0	0	341		
8x8	0	0	0	192	
4x4	0	0	0	0	0

Source: Frame from the sample file showing VP9 image blocks marked as skipped (Skip) using a VP9 analyzer.

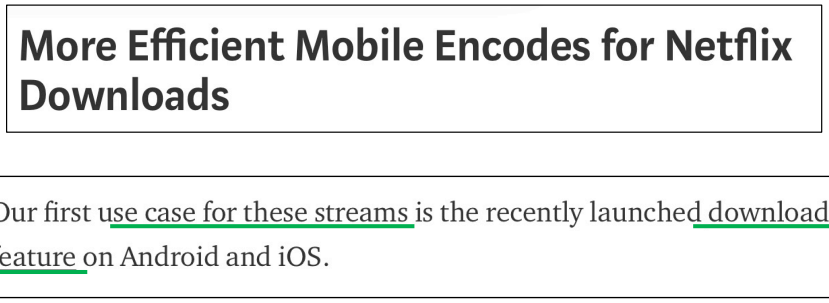
49. Below shows a frame from the sample file showing a block marked as skipped in Segment 0, referencing a previous frame (Reference 0) and a motion vector using a VP9 analyzer.





Source: Frame from the sample file showing a block marked as skipped in Segment 0, referencing a previous frame (Reference 0) and a motion vector using a VP9 analyzer.

50. The Accused Infringing Devices provide the ability to store the result of the video encoding to a file by creating downloadable coded data.



Source: https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909

51. The Accused Infringing Devices use INTER blocks marked as skipped (skip flag is true) to code a block without sending any coded pixel information for a macroblock in the current frame.

52. The decision to skip the coding of pixel information is based on a difference in pixel value for a macroblock between the current macroblock, and its corresponding macroblock in a previous frame. This difference is based on a test statistic between the macroblock in the current frame and a reference frame.

53. The Accused Infringing Devices perform this step by using test statistics such as variance, or the magnitude of the residual frequency components in macroblock to be coded or skipped, with comparisons to predetermined thresholds of zero for the variance, **ac_thr** for the AC coefficients and **dc_thr** for the DC coefficient.

```
616     // Check if all ac coefficients can be quantized to zero.
617     if (var_tx < ac_thr || var == 0) {
618       x->skip_txfm[0] = SKIP_TXFM_AC_ONLY;
619       // Check if dc coefficient can be quantized to zero.
620       if (sse_tx - var_tx < dc_thr || sse == var)
621         x->skip_txfm[0] = SKIP_TXFM_AC_DC;
622     } else {
623       if (sse_tx - var_tx < dc_thr || sse == var) skip_dc = 1;
624     }
```

Source:
https://chromium.googlesource.com/webm/libvpx/+/master/vp9/encoder/vp9_pickmode.c

54. Netflix has infringed, and continues to infringe, at least claim 1 of the '229 patent in the United States, by making, using, offering for sale, selling, licensing and/or importing the Accused Infringing Devices in violation of 35 U.S.C. § 271(a).

55. Upon information and belief, Netflix may have infringed and continues to infringe the '229 patent through other software and devices utilizing the same or reasonably similar functionality, including other versions of the Accused Infringing Devices.

56. Netflix's acts of direct infringement have caused and continue to cause damage to Uniloc and Uniloc is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

COUNT III – INFRINGEMENT OF U.S. PATENT NO. 6,519,005

57. The allegations of paragraphs 1-8 of this Complaint are incorporated by reference as though fully set forth herein.

58. The '005 patent, titled "Method of Concurrent Multiple-Mode Motion Estimation For Digital Video," issued on February 11, 2003. A copy of the '005 patent is attached as Exhibit C.

59. Pursuant to 35 U.S.C. § 282, the '005 patent is presumed valid.

60. Invented by Koninklijke Philips Electronics N.V., the inventions of the '005 patent were not well-understood, routine or conventional at the time of the invention. At the time of invention of the '005 patent, different compression algorithms had been developed for digitally encoding video and audio information (hereinafter referred to generically as "digital video data stream") in order to minimize the bandwidth required to transmit this digital video data stream for a given picture quality. '005 patent at 1:12-17. Several multimedia specification committees established and proposed standards for encoding/compressing and decoding/decompressing audio and video information. The most widely accepted international standards have been proposed by the Moving Pictures Expert Group (MPEG). *Id*. at 1:17-22 Video coding, such as MPEG coding, involves a number of steps. In general, in accordance with the MPEG standards, the audio and video data comprising a multimedia data stream (or "bit stream") are encoded/compressed in an intelligent manner using a compression technique generally known as "motion coding." *Id*. at 1:41-45. More particularly, rather than transmitting each video frame in its entirety, MPEG uses motion estimation for only those parts of sequential pictures that vary due to motion, where possible. *Id*. at 1:45-48. In general, the picture elements or "pixels" of a picture are specified relative to those of a previously transmitted reference or "anchor" picture using differential or "residual" video, as well as so-called "motion vectors" that specify the location of a 16-by-16 array of pixels or "macroblock" within the current picture relative to its original location within the anchor picture. *Id*. at 1:48-55. Computation of the motion vector(s) for a given macroblock involves an exhaustive search procedure

that is very computationally intensive. *Id*. at 3:25-39. It was desirable at the time of the invention to improve this process. *Id*. at 3:40-67.

61. The inventive solution of the claimed inventions of the '005 patent provides a system and method for digital video compression, and, more particularly, to a motion estimation method and search engine for a digital video encoder that is simpler, faster, and less expensive than prior art technology, and that permits concurrent motion estimation using multiple prediction modes. *Id*. at 1:6-11.

62. A person of ordinary skill in the art reading the '005 patent and its claims would understand that the patent's disclosure and claims are drawn to solving a specific, technical problem arising in the field of digital video compression. *Id*. Moreover, a person of ordinary skill in the art would understand that the claimed subject matter of the '005 patent presents advancements in the field of digital video compression, and more particularly to a motion estimation method and search engine for a digital video encoder that is simpler, faster, and less expensive than prior art technology, and that permits concurrent motion estimation using multiple prediction modes. *Id*.

63. In light of the foregoing, a person of ordinary skill in the art would understand that claim 1 of the '005 patent is directed to a method for motion coding an uncompressed digital video data stream, which provides concurrent motion estimation using multiple prediction modes. Moreover, a person of ordinary skill in the art would understand that claim 1 of the '005 patent contains that corresponding inventive concept.

64. On information and belief, Netflix makes, uses, offers for sale, sells in the United States, licenses in the United States and/or imports into the United States products and services such as its H.264 encoders that practice a method for motion coding an uncompressed digital video data stream (collectively the "Accused

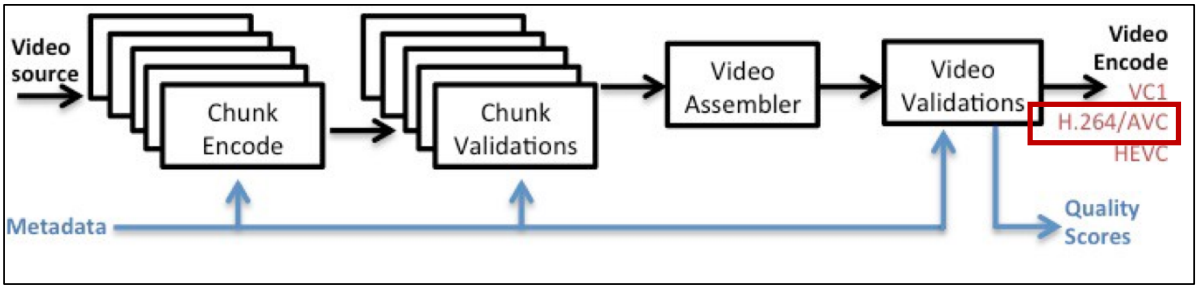
Infringing Devices").

65. Upon information and belief, the Accused Infringing Devices infringe at least claim 1 in the exemplary manner described below.

66. The Accused Infringing Devices provide a method for motion coding an uncompressed (pixel level) digital video data stream. The Accused Infringing Devices receive input video streams which are then encoded and/or transcoded using at least the H.264 standard. The H.264 standard is a widely used video compression format with decoder support on web browsers, TVs and other consumer devices. Moreover, H.264 uses motion compressor and estimator for motion coding video streams.

> **Compression Performance**
>
> In December 2016, we introduced AVCHi-Mobile and VP9-Mobile encodes for downloads. For these mobile encodes, several changes led to improved compression performance over per-title encodes, including longer GOPs, flexible encoder settings and per-chunk optimization. These streams serve as our high quality baseline for H.264/AVC and VP9 encoding with traditional rate control settings.

Source: https://medium.com/netflix-techblog/optimized-shot-based-encodes-now-streaming-4b9464204830



> parallel on different instances. For each chunk, bitstream-level and pixel-level analysis is applied to detect errors and generate metadata such as temporal and spatial fingerprints. After all the chunks are inspected, the results are

Source: https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-

d159db052746

0.6 Overview of the design characteristics

This subclause does not form an integral part of this Recommendation | International Standard.

The coded representation specified in the syntax is designed to enable a high compression capability for a desired image quality. With the exception of the transform bypass mode of operation for lossless coding in the High 4:4:4 Intra, CAVLC 4:4:4 Intra, and High 4:4:4 Predictive profiles, and the I_PCM mode of operation in all profiles, the algorithm is typically not lossless, as the exact source sample values are typically not preserved through the encoding and decoding processes. A number of techniques may be used to achieve highly efficient compression. Encoding algorithms (not specified in this Recommendation | International Standard) may select between inter and intra coding for block-shaped regions of each picture. Inter coding uses motion vectors for block-based inter prediction to exploit temporal statistical dependencies between different pictures. Intra coding uses various spatial prediction modes to exploit spatial statistical dependencies in the source signal for a single picture. Motion vectors and intra prediction modes may be specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform to remove spatial correlation inside the transform block before it is quantised, producing an irreversible process that typically discards less important visual information while forming a close approximation to the source samples. Finally, the motion vectors or intra prediction modes are combined with the quantised transform coefficient information and encoded using either variable length coding or arithmetic coding.

0.6.1 Predictive coding

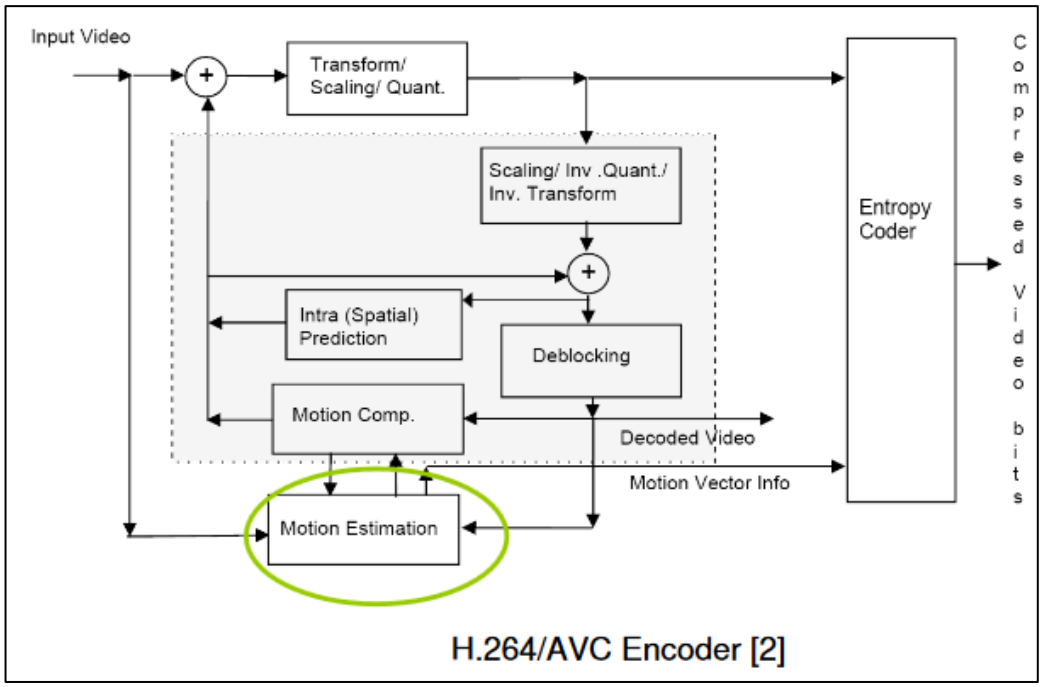
This subclause does not form an integral part of this Recommendation | International Standard.

Because of the conflicting requirements of random access and highly efficient compression, two main coding types are specified. Intra coding is done without reference to other pictures. Intra coding may provide access points to the coded sequence where decoding can begin and continue correctly, but typically also shows only moderate compression efficiency. Inter coding (predictive or bi-predictive) is more efficient using inter prediction of each block of sample values from some previously decoded picture selected by the encoder. In contrast to some other video coding standards, pictures coded using bi-predictive inter prediction may also be used as references for inter coding of other pictures.

The application of the three coding types to pictures in a sequence is flexible, and the order of the decoding process is generally not the same as the order of the source picture capture process in the encoder or the output order from the decoder for display. The choice is left to the encoder and will depend on the requirements of the application. The

decoding order is specified such that the decoding of pictures that use inter-picture prediction follows later in decoding order than other pictures that are referenced in the decoding process.

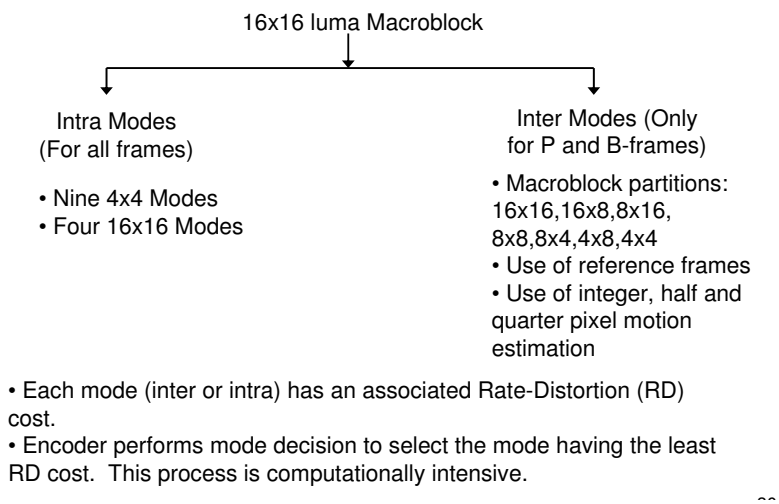
H.264 Encoder Block Diagram



Source: https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf

67. The Accused Infringing Devices provide a method for comparing pixels of a first pixel array (e.g., a macroblock) in a picture currently being coded with pixels of a plurality of second pixel arrays in at least one reference picture and concurrently performing motion estimation for each of a plurality of different prediction modes in order to determine which of the prediction modes is an optimum prediction mode.

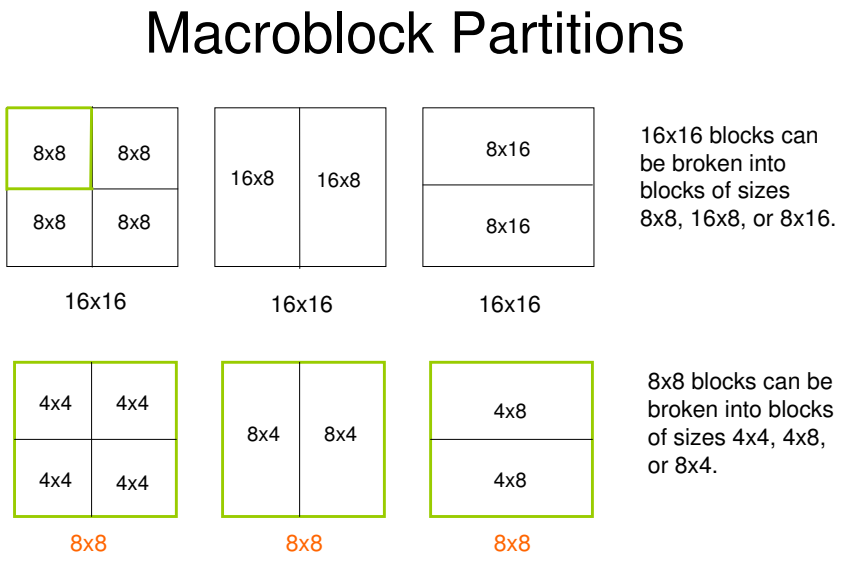
68. H.264 uses different motion estimation modes in inter-frame prediction. These modes are commonly referred to as inter-frame prediction modes, or inter modes. Each inter mode involves partitioning the current Macroblock into a different combination of sub blocks, and selecting the optimum motion vector for the current Macroblock based on the partition. The inter-frame prediction modes, or inter modes, can be further categorized by the number and position of the reference frames, as well as the choice of integer pixel, half pixel and quarter pixel values in motion estimation. The Netflix H.264 encoders concurrently perform motion estimation of a Macroblock for all inter-modes and select the most optimum prediction mode with least rate distortion cost.



Source: https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf, p. 30

69. H.264 provides a hierarchical way to partition a Macroblock, with the available partitions shown in the following two figures. An exemplary inter-frame prediction mode, or inter mode, can be for a Macroblock to be partitioned to encompass a 16x8 sub block on the left, and two 8x8 sub blocks on the right.

Macroblock partitions for inter-frame prediction modes



Source: https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf, p. 4

H.264 provides macroblock partitions for inter-frame prediction modes

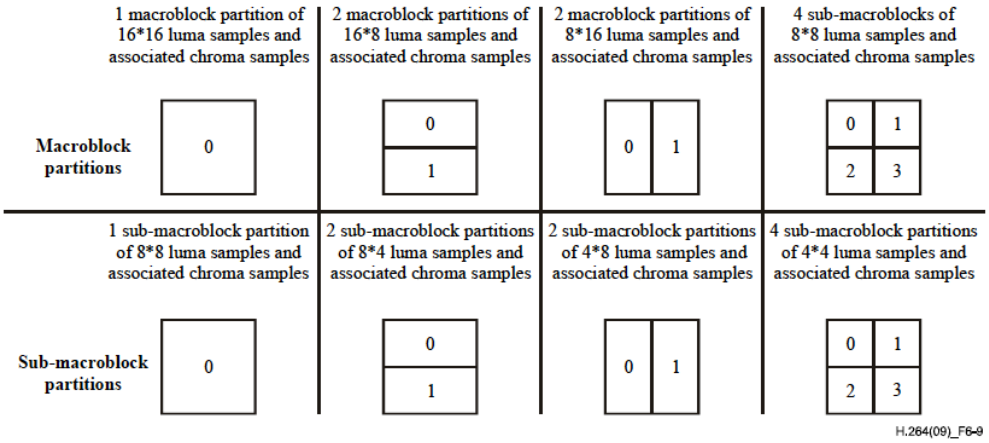


Figure 6-9 – Macroblock partitions, sub-macroblock partitions, macroblock partition scans, and sub-macroblock partition scans

Source: H.264 Standard (03-2010) at p. 26

70. The optimum prediction mode as chosen for the current Macroblock is embedded in the compressed bit stream of H.264, as shown in the following two syntaxes.

Macroblock prediction syntax in H.264

7.3.5.1 Macroblock prediction syntax

mb_pred(mb_type) {	C	Descriptor
if(MbPartPredMode(mb_type, 0) == Intra_4x4 || MbPartPredMode(mb_type, 0) == Intra_16x16) {		
if(MbPartPredMode(mb_type, 0) == Intra_4x4)		
for(luma4x4BlkIdx=0; luma4x4BlkIdx<16; luma4x4BlkIdx++) {		
prev_intra4x4_pred_mode_flag[luma4x4BlkIdx **]**	2	u(1) | ae(v)
if(!prev_intra4x4_pred_mode_flag[luma4x4BlkIdx])		
rem_intra4x4_pred_mode[luma4x4BlkIdx **]**	2	u(3) | ae(v)
}		
intra_chroma_pred_mode	2	ue(v) | ae(v)
} else if(MbPartPredMode(mb_type, 0) != Direct) {		
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 || mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L1)		
ref_idx_l0[mbPartIdx **]**	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 || mb_field_decoding_flag) && MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
ref_idx_l1[mbPartIdx **]**	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode (mb_type, mbPartIdx) != Pred_L1)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx **][** 0 **][** compIdx **]**	2	se(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < NumMbPart(mb_type); mbPartIdx++)		
if(MbPartPredMode(mb_type, mbPartIdx) != Pred_L0)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx **][** 0 **][** compIdx **]**	2	se(v) | ae(v)
}		
}		

Source: H.264 Standard (03-2010) at p. 57

Sub-macroblock prediction syntax in H.264

7.3.5.2 Sub-macroblock prediction syntax

sub_mb_pred(mb_type) {	C	Descriptor
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
sub_mb_type[mbPartIdx]	2	ue(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l0_active_minus1 > 0 || mb_field_decoding_flag) && mb_type != P_8x8ref0 && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
ref_idx_l0[mbPartIdx]	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if((num_ref_idx_l1_active_minus1 > 0 || mb_field_decoding_flag) && sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
ref_idx_l1[mbPartIdx]	2	te(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L1)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l0[mbPartIdx][subMbPartIdx][compIdx]	2	se(v) | ae(v)
for(mbPartIdx = 0; mbPartIdx < 4; mbPartIdx++)		
if(sub_mb_type[mbPartIdx] != B_Direct_8x8 && SubMbPredMode(sub_mb_type[mbPartIdx]) != Pred_L0)		
for(subMbPartIdx = 0; subMbPartIdx < NumSubMbPart(sub_mb_type[mbPartIdx]); subMbPartIdx++)		
for(compIdx = 0; compIdx < 2; compIdx++)		
mvd_l1[mbPartIdx][subMbPartIdx][compIdx]	2	se(v) | ae(v)
}		

Source: H.264 Standard (03-2010) at p. 58

71. The Accused Infringing Devices provide a method for determining which of the second pixel arrays (e.g., macroblock) constitutes a best match with respect to the first pixel array (e.g., macroblock) for the optimum prediction mode.

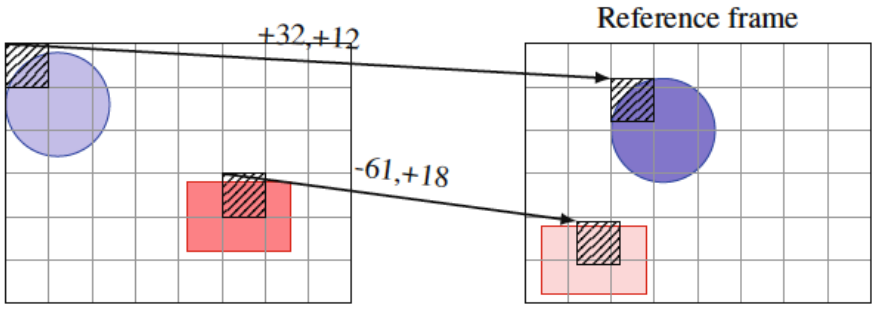


Fig. 2.4: Motion estimation. For each MB the best matching block in the reference frame is found. The encoder codes the differences (errors) between the MBs and their best matching blocks. Arrows indicate motion vectors and are labeled by the vector coordinates. In this example the shapes are identical but their colors are slightly larger/darker.

Source: B. Juurlink et al., Scalable Parallel Programming Applied to H.264, Chapter 2: Understanding the Application: An Overview of the H.264 Standard, p. 12

72. For example, the encoder performs mode decision to select the most optimum prediction mode with least rate distortion cost.

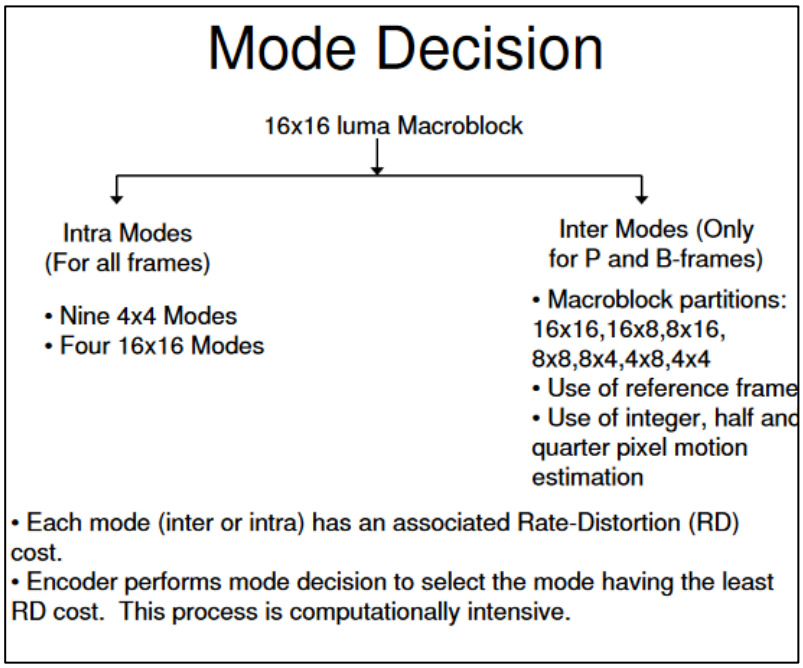
Macroblock layer semantics

The following semantics are assigned to the macroblock types in Table 7-13:

- P_L0_16x16: the samples of the macroblock are predicted with one luma macroblock partition of size 16x16 luma samples and associated chroma samples.
- P_L0_L0_MxN, with MxN being replaced by 16x8 or 8x16: the samples of the macroblock are predicted using two luma partitions of size MxN equal to 16x8, or two luma partitions of size MxN equal to 8x16, and associated chroma samples, respectively.
- P_8x8: for each sub-macroblock an additional syntax element (sub_mb_type[mbPartIdx] with mbPartIdx being the macroblock partition index for the corresponding sub-macroblock) is present in the bitstream that specifies the type of the corresponding sub-macroblock (see subclause 7.4.5.2).
- P_8x8ref0: has the same semantics as P_8x8 but no syntax element for the reference index (ref_idx_l0[mbPartIdx] with mbPartIdx = 0..3) is present in the bitstream and ref_idx_l0[mbPartIdx] shall be inferred to be equal to 0 for all sub-macroblocks of the macroblock (with indices mbPartIdx = 0..3).
- P_Skip: no further data is present for the macroblock in the bitstream.

Source: H.264 Standard (03-2010), p. 100

Mode Decision



Source: https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf, p. 30

73. The Accused Infringing Devices provide a method for generating a motion vector for the first pixel array in response to the determining step. The encoder calculates the appropriate motion vectors and other data elements represented in the video data stream.

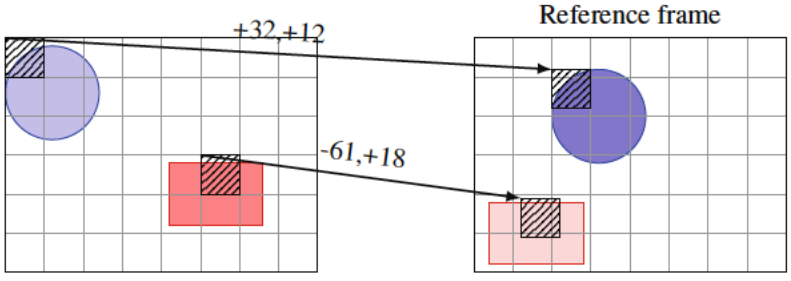


Fig. 2.4: Motion estimation. For each MB the best matching block in the reference frame is found. The encoder codes the differences (errors) between the MBs and their best matching blocks. Arrows indicate motion vectors and are labeled by the vector coordinates. In this example the shapes are identical but their colors are slightly larger/darker.

Source: B. Juurlink et al., Scalable Parallel Programming Applied to H.264, Chapter 2: Understanding the Application: An Overview of the H.264 Standard, p. 12

Motion Vector Derivation is described below

1. The derivation process for motion vector components and reference indices as specified in subclause 8.4.1 is invoked.

   Inputs to this process are:

   – a macroblock partition mbPartIdx,

   – a sub-macroblock partition subMbPartIdx.

   Outputs of this process are:

   – luma motion vectors mvL0 and mvL1 and when ChromaArrayType is not equal to 0, the chroma motion vectors mvCL0 and mvCL1

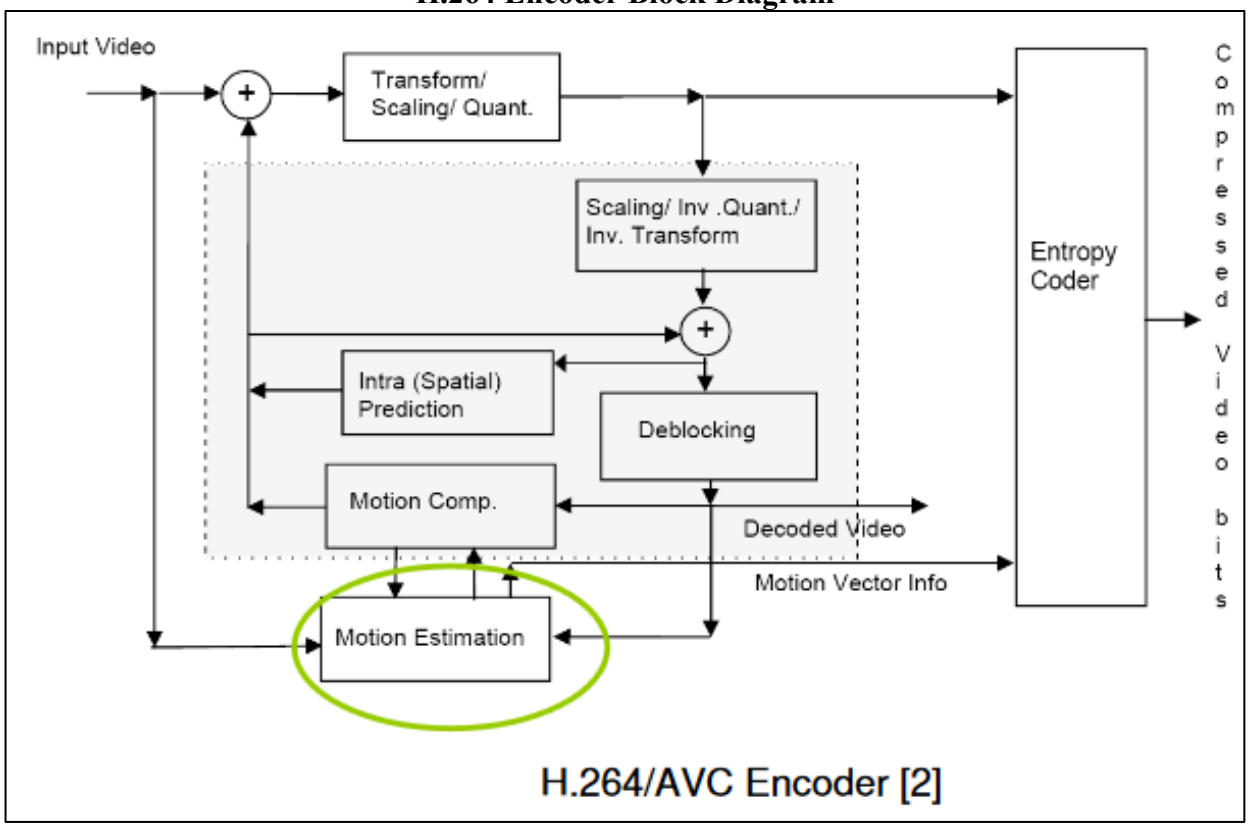
   – reference indices refIdxL0 and refIdxL1

   – prediction list utilization flags predFlagL0 and predFlagL1

   – the sub-macroblock partition motion vector count subMvCnt.

Source: H.264 Standard (03-2010), p. 151

H.264 Encoder Block Diagram



Source: https://courses.cs.washington.edu/courses/csep590a/07au/lectures/rahullarge.pdf, p. 2

74. Netflix has infringed, and continues to infringe, at least claim 1 of the '005 patent in the United States, by making, using, offering for sale, selling, licensing and/or importing the Accused Infringing Devices in violation of 35 U.S.C.

§ 271(a).

75. Upon information and belief, Netflix may have infringed and continues to infringe the '005 patent through other software and devices utilizing the same or reasonably similar functionality, including other versions of the Accused Infringing Devices.

76. Netflix's acts of direct infringement have caused and continue to cause damage to Uniloc and Uniloc is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

PRAYER FOR RELIEF

WHEREFORE, plaintiff Uniloc 2017 LLC respectfully prays that the Court enter judgment in their favor and against Netflix as follows:

a. A judgment that Netflix has infringed one or more claims of the '609 Patent literally and/or under the doctrine of equivalents;

b. A judgment that Netflix has infringed one or more claims of the '229 Patent literally and/or under the doctrine of equivalents;

c. A judgment that Netflix has infringed one or more claims of the '005 Patent literally and/or under the doctrine of equivalents;

d. That for each Asserted Patent this Court judges infringed by Netflix this Court award Uniloc its damages pursuant to 35 U.S.C. § 284 and any royalties determined to be appropriate;

e. That this be determined to be an exceptional case under 35 U.S.C. § 285 and that Uniloc be awarded enhanced damages up to treble damages for willful infringement as provided by 35 U.S.C. § 284;

f. That this Court award Uniloc prejudgment and post-judgment interest on its damages;

g. That Uniloc be granted its reasonable attorneys' fees in this action;

h. That this Court award Uniloc its costs; and

i. That this Court award Uniloc such other and further relief as the Court deems proper.

DEMAND FOR JURY TRIAL

Uniloc hereby demands trial by jury on all issues so triable pursuant to Fed. R. Civ. P. 38.

Dated: November 17, 2018

FEINBERG DAY ALBERTI LIM & BELLOLI LLP

By: */s/ M. Elizabeth Day*
M. Elizabeth Day

Attorneys for Plaintiff
Uniloc 2017 LLC